AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing Of Claims:

1. (Original) A method for performing an extended exponent floating point operation on a plurality of operands to produce a product of the plurality of operands, the method comprising:

grouping the plurality of operands into at least one group;

determining a plurality of scale factors for the plurality of operands, respectively, and providing a running sum of the plurality of scale factors;

scaling the plurality of operands to obtain a plurality of scaled operands;

multiplying the plurality of scaled operands to obtain a group product and scaling the group product to obtain a scaled group product; and

adjusting the scaled group product based on the running sum,

wherein the plurality of operands are grouped such that when all the plurality of scaled operands in the at least one group are multiplied an overflow or underflow will not occur.

2. (Original) The method according to claim 1, wherein at least one of the plurality of scaled operands is greater than or equal to one and less than two.

3. (Original) The method according to claim 1, wherein determining the plurality of scale factors comprises determining at least one scale factor of the plurality of scale factors based on a format of at least one corresponding operand of the plurality of operands.

4. (Original) The method according to claim 3, wherein determining at least one scale factor of the plurality of scale factors comprises:

determining the format of the corresponding operand of the plurality of operands;

setting the at least one scale factor to zero if the format is a zero format, an underflow format, an overflow format, an infinity format, or a NaN format;

setting the at least one scale factor based on exponent field bits of the corresponding operand if the format is a normalized non-zero format;

setting the at least one scale factor based on a number of leading zeros in a fraction field of the corresponding operand if the format is a denormalized format; and

setting the at least one scale factor based on a number of trailing zeros in the fraction field of the corresponding operand if the format is a delimited format.

5. (Original) The method according to claim 1, wherein at least one operand of the plurality of operands is scaled based on a format of the at least one operand.

6. (Original) The method according to claim 5, wherein scaling the at least one operand comprises:

determining the format of the at least one operand;

setting exponent field bits of the at least one operand to a predetermined bit pattern if the format of the at least one operand is one of the group of a normalized non-zero format, a denormalized format, and a delimited format;

maintaining the exponent field bits and fraction field bits of the at least one operand if the format of the at least one operand is a zero format, an underflow format, an overflow format, an infinity format, or a NaN format;

normalizing the fraction field bits of the at least one operand if the format of the at least one operand is a denormalized format; and

maintaining the fraction field bits of the at least one operand, except a delimiter bit, which is set equal to zero, if the format of the at least one operand is a delimited format.

7. (Original) The method according to claim 1, wherein at least one operand of the plurality of operands stores status information wherein the status information is preserved in the product.

8. (Original) A computer-readable medium on which is stored a set of instructions for performing an extended exponent floating point operation on a plurality of operands to produce a product of the plurality of operands, which when executed performs steps comprising:

grouping the plurality of operands into at least one group;

determining a plurality of scale factors for the plurality of operands, respectively, and providing a running sum of the plurality of scale factors;

scaling the plurality of operands to obtain a plurality of scaled operands;

multiplying the plurality of scaled operands to obtain a group product and scaling the group product to obtain a scaled group product; and

adjusting the scaled group product based on the running sum,

wherein the plurality of operands are grouped such that when all the plurality of scaled operands in the at least one group are multiplied an overflow or underflow will not occur.

9. (Original) The computer-readable medium according to claim 8, wherein at least one of the plurality of scaled operands is greater than or equal to one and less than two.

10. (Original) The computer-readable medium according to claim 8, wherein determining the plurality of scale factors comprises determining at least one scale factor of the plurality of scale factors based on a format of at least one corresponding operand of the plurality of operands.

11. (Original) The computer-readable medium according to claim 10, wherein determining at least one scale factor of the plurality of scale factors comprises:

determining the format of the corresponding operand of the plurality of operands;

setting the at least one scale factor to zero if the format is a zero format, an underflow format, an overflow format, an infinity format, or a NaN format;

setting the at least one scale factor based on exponent field bits of the corresponding operand if the format is a normalized non-zero format;

setting the at least one scale factor based on a number of leading zeros in a fraction field of the corresponding operand if the format is a denormalized format; and

setting the at least one scale factor based on a number of trailing zeros in the fraction field of the corresponding operand if the format is a delimited format.

12. (Original) The computer-readable medium according to claim 8, wherein at least one operand of the plurality of operands is scaled based on a format of the at least one operand.

13. (Original) The computer-readable medium according to claim 12, wherein scaling the at least one operand comprises:

determining the format of the at least one operand;

setting exponent field bits of the at least one operand to a predetermined bit pattern if the format of the at least one operand is one of the group of a normalized non-zero format, a denormalized format, and a delimited format;

maintaining the exponent field bits and fraction field bits of the at least one operand if the format of the at least one operand is a zero format, an underflow format, an overflow format, an infinity format, or a NaN format;

normalizing the fraction field bits of the at least one operand if the format of the at least one operand is a denormalized format; and

maintaining the fraction field bits of the at least one operand, except a delimiter bit, which is set equal to zero, if the format of the at least one operand is a delimited format.

14. (Original) The computer-readable medium according to claim 8, wherein at least one operand of the plurality of operands stores status information wherein the status information is preserved in the product.

15. (Currently Amended) A system for performing an extended exponent floating point operation on a plurality of operands to produce a product of the plurality of operands, the system comprising:

a running sum unit for determining a corresponding scale factor for each of the plurality of operands based on a format of the operand and computing a running sum of the scale factors;

a scaling unit for scaling each of the plurality of operands to obtain a plurality of scaled operands based on a format of the operand and computing a running product of the scaled plurality of operands; and

a control module for receiving the running sum from the running sum unit and the running product from the scaling unit and adjusting the running product based on the running sum, wherein the control module provides the plurality of operands ~~and the running sum~~ to the running sum unit and ~~the plurality of operands and the running product~~ to the scaling unit and wherein the control module provides the running sum calculated in a previous iteration to the running sum unit during a subsequent iteration

and wherein the control module provides the running product calculated in the previous iteration to the scaling unit during the subsequent iteration.

16. (Original) The system according to claim 15, wherein at least one operand of the plurality of operands stores status information and the scaling unit preserves the status information in the corresponding scaled operand.

17. (Original) The system according to claim 15, wherein the running sum unit comprises:

an operand analysis circuit for determining for each of the plurality of operands a corresponding format; and

a processing circuit for determining for each of the plurality of operands a corresponding scale factor based on the corresponding format.

18. (Original) The system according to claim 15, wherein the scaling unit comprises:

an operand analysis circuit for determining for each of the plurality of operands a corresponding format; and

a processing circuit for scaling each of the plurality of operands based on the corresponding format.

19. (Original) A method for scaling a floating point operand to obtain a scaled floating point operand comprising:

determining a format of the floating point operand;

setting exponent field bits of the floating point operand to a predetermined bit pattern if the format of the floating point operand is one of the group of a normalized non-zero format, a denormalized format, and a delimited format;

maintaining the exponent field bits and fraction field bits of the floating point operand if the format of the floating point operand is a zero format, an underflow format, an overflow format, an infinity format, or a NaN format;

normalizing the fraction field bits of the floating point operand if the format of the floating point operand is a denormalized format; and

maintaining the fraction field bits of the floating point operand, except a delimiter bit, which is set equal to zero, if the format of the floating point operand is a delimited format.

20. (Original) The method according to claim 19, wherein the floating point operand stores status information wherein the status information is preserved in the scale floating point operand.

21. (Original) A method for determining a scale factor of a floating point operand wherein when the floating point operand is multiplied by the scale factor the resulting scaled floating point operand is within a predetermined range, the method comprising:

determining a format of the floating point operand;

setting the scale factor to zero if the format is a zero format, an underflow format, an overflow format, an infinity format, or a NaN format;

setting the scale factor based on exponent field bits of the floating point operand if the format is a normalized non-zero format;

setting the scale factor based on a number of leading zeros in a fraction field of the floating point operand if the format is a denormalized format; and

setting the scale factor based on a number of trailing zeros in the fraction field of the floating point operand if the format is a delimited format.

22. (Original) A computer-readable medium on which is stored a set of instructions for scaling a floating point operand to obtain a scaled floating point operand, which when executed performs steps comprising:

determining a format of the floating point operand;

setting exponent field bits of the floating point operand to a predetermined bit pattern if the format of the floating point operand is one of the group of a normalized non-zero format, a denormalized format, and a delimited format;

maintaining the exponent field bits and fraction field bits of the floating point operand if the format of the floating point operand is a zero format, an underflow format, an overflow format, an infinity format, or a NaN format;

normalizing the fraction field bits of the floating point operand if the format of the floating point operand is a denormalized format; and

maintaining the fraction field bits of the floating point operand, except a delimiter bit, which is set equal to zero, if the format of the floating point operand is a delimited format.

23. (Original) A computer-readable medium on which is stored a set of instructions for determining a scale factor of a floating point operand wherein when the floating point operand is multiplied by the scale factor the resulting scaled floating point operand is within a predetermined range, which when executed performs steps comprising:

determining a format of the floating point operand;

setting the scale factor to zero if the format is a zero format, an underflow format, an overflow format, an infinity format, or a NaN format;

setting the scale factor based on exponent field bits of the floating point operand if the format is a normalized non-zero format;

setting the scale factor based on a number of leading zeros in a fraction field of the floating point operand if the format is a denormalized format; and

setting the scale factor based on a number of trailing zeros in the fraction field of the floating point operand if the format is a delimited format.

24. (Currently Amended) An apparatus for scaling a floating point operand to obtain a scaled floating point operand comprising:

an operand analysis circuit for determining a format of the floating point operand; and

a processing circuit for scaling ~~each of~~ the floating point operand based on the format.

25. (Currently Amended) An apparatus for determining a scale factor of a floating point operand wherein when the floating point operand is multiplied by the scale factor the resulting scaled floating point operand is within a predetermined range, the apparatus comprising:

an operand analysis circuit ~~for determining~~ for determining a format of the floating point operand; and

a processing circuit for determining the scale factor based on the format.